Minimax-optimal and Locally-Adaptive Online Nonparametric Regression

Séminaire du SAMM, Paris, France

Paul Liautaud

April 25, 2025

Sorbonne Université, LPSM, Paris

Joint work with

Pierre Gaillard
CR Inria/UGA

Olivier Wintenberger
PR LPSM/SU

Table of contents

From *statistical* to *online* learning

Classical Machine Learning

The learner: 

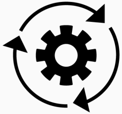
❶ observes a **whole training dataset** with labels/targets:

$$(x_1, y_1), \ldots, (x_T, y_T) \stackrel{\text{iid}}{\sim} (X, Y) \text{ with distribution } \mathbb{P} \text{ over } \mathcal{X} \times \mathcal{Y}.$$

Classical Machine Learning

The learner:

$\longrightarrow$



❶ observes a **whole training dataset** with labels/targets:

$$(x_1, y_1), \ldots, (x_T, y_T) \overset{\text{iid}}{\sim} (X, Y) \text{ with distribution } \mathbb{P} \text{ over } \mathcal{X} \times \mathcal{Y}.$$

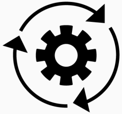
❷ builds a function $\widehat{f} : \mathcal{X} \to \mathcal{Y} \in \mathcal{F}$ with small risk $\mathbb{E}_{\mathbb{P}}[\ell(\widehat{f}(X), Y)]$ by minimizing:

$$R(\widehat{f}) = \frac{1}{T} \sum_{t=1}^{T} \ell(\widehat{f}(x_t), y_t),$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a prescribed loss function.

Classical Machine Learning

The learner: 

$\longrightarrow$  $\longrightarrow$ 

❶ observes a **whole training dataset** with labels/targets:

$$(x_1, y_1), \ldots, (x_T, y_T) \overset{\text{iid}}{\sim} (X, Y) \text{ with distribution } \mathbb{P} \text{ over } \mathcal{X} \times \mathcal{Y}.$$

❷ builds a function $\widehat{f} : \mathcal{X} \to \mathcal{Y} \in \mathcal{F}$ with small risk $\mathbb{E}_{\mathbb{P}}[\ell(\widehat{f}(X), Y)]$ by minimizing:

$$R(\widehat{f}) = \frac{1}{T} \sum_{t=1}^{T} \ell(\widehat{f}(x_t), y_t),$$

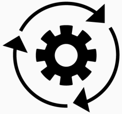
where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a prescribed loss function.

❸ controls the error of new data if they are similar to the training data.

Classical Machine Learning

The learner:

$\longrightarrow$



$\longrightarrow$

❶ observes a **whole training dataset** with labels/targets:

$$(x_1, y_1), \ldots, (x_T, y_T) \overset{\text{iid}}{\sim} (X, Y) \text{ with distribution } \mathbb{P} \text{ over } \mathcal{X} \times \mathcal{Y}.$$

❷ builds a function $\widehat{f} : \mathcal{X} \to \mathcal{Y} \in \mathcal{F}$ with small risk $\mathbb{E}_{\mathbb{P}}[\ell(\widehat{f}(X), Y)]$ by minimizing:

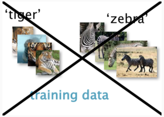
$$R(\widehat{f}) = \frac{1}{T} \sum_{t=1}^{T} \ell(\widehat{f}(x_t), y_t),$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a prescribed loss function.

❸ controls the error of new data if they are similar to the training data.

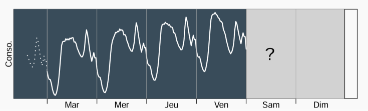
✗ We won't deal with it! ✗

In **sequential learning**:
- data are acquired and treated **on the fly**;
- data are **not** necessarily **iid**, possibly **adversarial**;
- feedbacks are received and algorithms updated **step by step**.



? Why **online** learning? In some applications, the environment may **evolve over time** and data may be available **sequentially**, e.g.:

- ads to display,
- electricity consumption forecast,
- spam detection,
- aggregation of expert knowledge.



Setting: online regression with individual sequences (1/2)

Online prediction scenario: at each round $t \in \mathbb{N}^*$, the forecaster

1. observes an input $x_t \in \mathcal{X}$;
2. chooses a prediction $\widehat{f}_t(x_t) \in \mathbb{R}$; — Choose $\widehat{f}_t$ before observing ℓ_t
3. incurs a loss $\ell_t(\widehat{f}_t(x_t))$ — No assumptions on how ℓ_t is generated
4. updates his prediction function $\widehat{f}_t \to \widehat{f}_{t+1}$ — Based on observed gradients

$\cdots \to x_t =$  $\to$  $\to$ 'zebra' $\to$  $\to x_{t+1} =$  $\to \cdots$

Goal: given some large (nonparametric) function set $\mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$, we want to minimize the regret against any competitor $f \in \mathcal{F}$

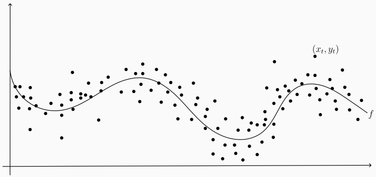
$$\mathrm{Reg}_T(f) = \underbrace{\sum_{t=1}^{T} \ell_t(\widehat{f}_t(x_t))}_{\text{our performance}} - \underbrace{\sum_{t=1}^{T} \ell_t(f(x_t))}_{\text{reference performance}} \underbrace{= o(T)}_{\text{goal}} .$$

Setting: online regression with individual sequences (2/2)

Individual sequences: no stochastic assumption on data (x_t, ℓ_t)!
$\widehat{f}_1, \ldots, \widehat{f}_T$ have to perform **well** with all **arbitrary** and possibly **adversarial** sequences.

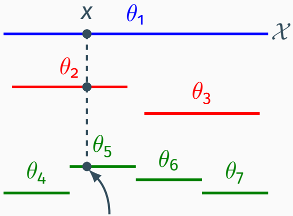
Assumptions:

- $\ell_1, \ldots, \ell_T$ are G-Lipschitz convex losses, with $G > 0$;
- $\mathcal{X} \subset \mathbb{R}^d$ bounded compact subset;
- $\mathcal{F} \subset [-B, B]^{\mathcal{X}}$ for some $B > 0$;
- $\mathcal{F} \subset \mathscr{C}^\alpha(L)$ the set of α-Hölder continuous functions, with $\alpha \in (0, 1]$, $L > 0$ **unknown**.



Parameter-free online approach with chaining trees

Chaining tree of depth $M = 3$:



$\mathtt{path}_{\mathcal{T}}(x) = \{1, 2, 5\}$
$\widehat{f}(x) = \theta_1 + \theta_2 + \theta_5$

Definition - *Chaining tree*

A Chaining-Tree (CT) prediction function $\widehat{f}$ over $\mathcal{X}$ is defined as:

$$\widehat{f}(x) = \sum_{n \in \mathcal{N}(\mathcal{T})} \theta_n \mathbf{1}_{x \in \mathcal{X}_n}, \quad x \in \mathcal{X}$$

where each interior node $n \in \mathcal{N}(\mathcal{T}) \backslash \mathcal{L}(\mathcal{T})$ has 2^d children forming a regular partition of $\mathcal{X}_n$.

Remark: contrary to standard methods, we predict with **all** nodes $n \in \mathcal{N}(\mathcal{T})$.

Parameter-free online algorithm

Algorithm 1: Training CT $\mathcal{T}$ at time $t \geqslant 1$

Input: $(\theta_{n,t})_{n\in\mathcal{N}(\mathcal{T})}$ (node predictors of $\mathcal{T}$), $(g_{n,t})_{n\in\mathcal{N}(\mathcal{T})}$ (gradients - later specified).

for $n \in \mathcal{N}(\mathcal{T})$ **do**

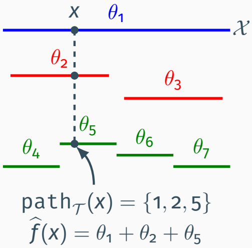
Predict $\widehat{f}_t(x_t) = \sum_{n\in\mathcal{N}(\mathcal{T})} \theta_{n,t}\mathbf{1}_{x_t\in\mathcal{X}_n}$;

Find $\theta_{n,t+1} \in \mathbb{R}$ to approximately minimize

$$\theta_n \mapsto \ell_t(\widehat{f}_{-n,t}(x_t)+\theta_n\mathbf{1}_{x_t\in\mathcal{X}_n}) \quad \text{with} \quad \widehat{f}_{-n,t}(x_t) = \widehat{f}_t(x_t)-\theta_{n,t}\mathbf{1}_{x_t\in\mathcal{X}_n} \tag{1}$$

using gradient $g_{n,t} = \left[\frac{\partial \ell_t\left(\widehat{f}_{-n,t}(x_t)+\theta_n\mathbf{1}_{x_t\in\mathcal{X}_n}\right)}{\partial\theta_n}\right]_{\theta_n=\theta_{n,t}}$.

Output: $(\theta_{n,t+1})_{n\in\mathcal{N}(\mathcal{T})}$



Our algorithm is **computationally tractable**

First result: global minimax-optimal regret against $\mathscr{C}^\alpha(L)$

Assumption - *Parameter free*

For any $n \in \mathcal{N}(\mathcal{T})$ and $\theta_n \in \mathbb{R}$, $\quad \sum_{t=1}^{T} g_t(\theta_{n,t} - \theta_n) \lesssim |\theta_n| \sqrt{\sum_{t=1}^{T} |g_{n,t}|^2}$.

After $T \geqslant 1$ rounds, Alg. 1 achieves a regret bounded as:

$$\sup_{f \in \mathscr{C}^\alpha(L)} \mathrm{Reg}_T(f) \lesssim GB\sqrt{T} + GL(f) \begin{cases} \sqrt{T}, & \text{if } d < 2\alpha, \\ \log_2 T\sqrt{T}, & \text{if } d = 2\alpha, \\ T^{1-\frac{\alpha}{d}}, & \text{if } d > 2\alpha. \end{cases}$$

★ **Adaptivity** to both α and $L(f) := \sup_{x \neq y} \frac{|f(x)-f(y)|}{\|x-y\|^\alpha} \leqslant L$!

✔ Our rates are **minimax** over $\mathscr{C}^\alpha(L)$ for general convex losses (Rakhlin et al., 2015)

💻 Our algorithm is **computationally tractable**: we update $O(\frac{1}{d}\log_2(T))$ parameters at each round.

Main intuitions behind our algorithm

Decompose regret: $\mathrm{Reg}_T(f) = \underbrace{\sum_{t=1}^T \ell_t(\widehat{f}_t(x_t)) - \ell_t(\widehat{f}_M(x_t))}_{R_1\text{:estimation regret}} + \underbrace{\sum_{t=1}^T \ell_t(\widehat{f}_M(x_t)) - \ell_t(f(x_t))}_{R_2\text{:approximation regret}}$

Multi-scale approximation process of a chaining tree $\widehat{f}_M$:

❶ *Control of the coefficient decay:*

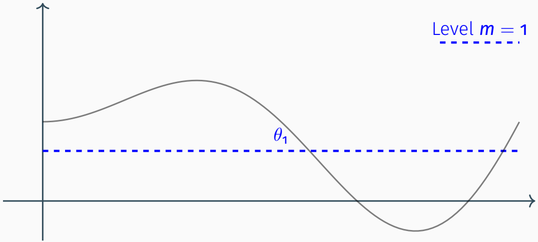
$$|\theta_{\text{level } m}| \leqslant L(f)2^{-\alpha m}$$

❷ *Control of estimation regret $(\widehat{f}_t) \to \widehat{f}_M$:*

$$R_1 \leqslant GL(f) \sum_{m=1}^{M} 2^{-\alpha m} \sqrt{2^{dm} T}\,.$$

❸ *Control of approximation regret:*

$$R_2 \leqslant GT \cdot \sup_{f \in \mathscr{C}^\alpha(L)} \|\widehat{f}_M - f\|_\infty \lesssim GTL(f)2^{-\alpha M}$$



Previous works: Gaillard and Gerchinovitz; Cesa-Bianchi et al. (2015; 2017) designed explicit chaining algorithms for square and absolute loss.

Generalisation to $\mathscr{C}^\alpha, \alpha \geqslant 1$ (1/2)

We use a predictor of the form

$$\widehat{f}_J = \bar{f} + \sum_{j=0}^{J} \sum_{k \in \Lambda_j} f_{j,k} \quad \text{with} \quad |\Lambda_j| = O(2^{jd}) \quad \text{for} \quad j \geqslant 0,$$

where $\bar{f}$ is a coarse approximation of f and $(f_{j,k})$ approaches f at finer scales.

Orthonormal Wavelet Basis: let $\{\psi_{j,k} : k \in \Lambda_j, j \geqslant -1\}$ an orthonormal $s > \lfloor \alpha \rfloor$-regular wavelet basis of $L^2(\mathcal{X})$ and

$$\bar{f} = \sum_{k \in \Lambda_{-1}} c_{-1,k} \psi_{-1,k} \quad \text{and} \quad f_{j,k} = c_{j,k} \psi_{j,k} \quad \text{for} \quad j \geqslant 0.$$

Control decay:

$$f \in \mathscr{C}^\alpha(L) \implies |c_{j,k}| \lesssim L(f) 2^{-\alpha m} \quad \text{for every } j \geqslant 0$$

Generalisation to $\mathscr{C}^\alpha, \alpha \geqslant 1$ (2/2)

Lauching Algorithm 1 on $\{(c_{j,k}) : k \in \Lambda_j, j \geqslant -1, \}$ over $T \geqslant 1$ rounds entails a regret, for every $\alpha > 0$

$$\sup_{f \in \mathscr{C}^\alpha(L)} \mathrm{Reg}_T(f) \lesssim GB|\Lambda_{-1}|\|\psi_{-1}\|_1\sqrt{T} + GL(f)\|\psi\|_2 \begin{cases} \sqrt{T}, & \text{if } d < 2\alpha, \\ \log_2 T\sqrt{T}, & \text{if } d = 2\alpha, \\ T^{1-\frac{\alpha}{d}}, & \text{if } d > 2\alpha. \end{cases}$$

- **Adaptivity** to both $L(f) \leqslant L$ and all $\alpha > 0$!
- Our rates are **minimax** over $\mathscr{C}^\alpha(L)$ for general convex losses (Rakhlin et al., 2015)

Another interesting result: Alg. 1 beats global adaptive OCO!

Comparison with global adaptive OCO:

- **Standard OCO:** updates a single global vector $\boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{N}(\mathcal{T})|}$ given a global gradient $\mathbf{g}_t$
- **Our method:** performs **node-wise updates** — each node n has its own parameter θ_n
  Localized gradients: $g_{n,t} = 0$ if $x_t \notin \mathcal{X}_n$
- **Result:** sparse, efficient updates with better regret bounds

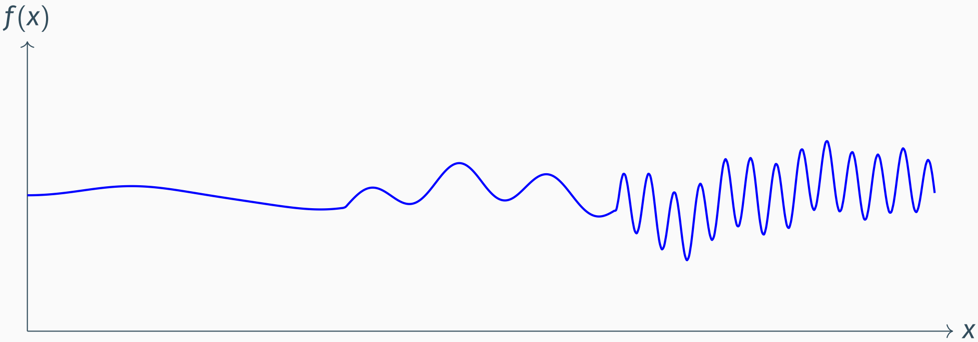
Regret comparison & key takeaway: when $p \leqslant 2$, our Algorithm 1 consistently achieves a lower regret than *any* global adaptive OMD method (e.g., adaptive OGD or EG).

Alg. 1: $O\left(\sum_n |\theta_n| \sqrt{\sum_t |g_{n,t}|^2}\right) \leqslant$ **Global OMD:** $O\left(\|\boldsymbol{\theta}\|_p \sqrt{\sum_t \|\mathbf{g}_t\|_q^2}\right)$

Locally adaptive algorithm

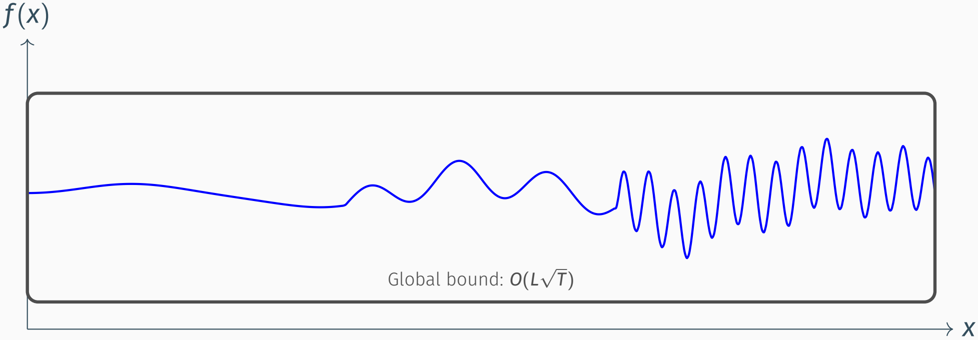
Motivation: why local adaptivity?

Idea: functions contain smooth and rough parts → we want to exploit **local smoothness**.



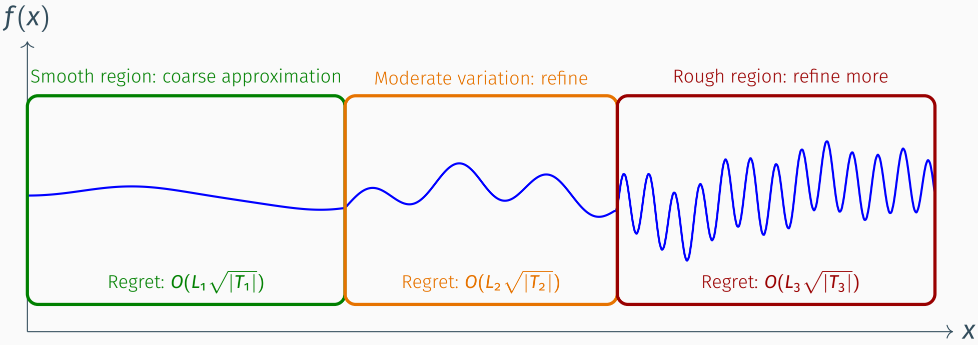
Motivation: why local adaptivity?

Idea: functions contain smooth and rough parts $\rightarrow$ we want to exploit **local smoothness**.



Motivation: why local adaptivity?

Idea: functions contain smooth and rough parts → we want to exploit **local smoothness**.



Result: instead of $O(L\sqrt{T})$, locally adaptive methods achieve $O\left(\sum_n L_n\sqrt{|T_n|}\right)$

We base our predictions on a *core tree* $\mathcal{T}_0$ partitionning $\mathcal{X}$ in $(\mathcal{X}_n)$ with

$$\widehat{f}_t(x_t) = \sum_{n \in \mathcal{N}(\mathcal{T}_0)} w_{n,t} \widehat{f}_{n,t}(x_t), \quad \text{for any } t \geqslant 1,$$

where:

- each $\widehat{f}_n$ is a local chaining-tree predictor over $\mathcal{X}_n \subset \mathcal{X}$,
- $(w_{n,t})$ are trainable parameters such that $\sum_n w_{n,t} = 1$.

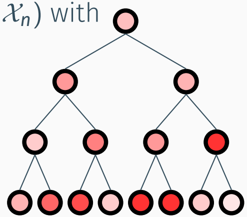


Figure 1: Example of $\mathcal{T}_0$

Expert aggregation procedure: train weights $\mathbf{w}_t = (w_{n,t})$ using gradients $\mathbf{g}_t = \nabla_{\mathbf{w}} \ell_t(\widehat{f}_t(x_t))|_{\mathbf{w}=\mathbf{w}_t}$ with any subroutine that satisfies:

Assumption - *Second-order algorithm*

For any $n \in \mathcal{N}(\mathcal{T}_0)$, $\quad \sum_{t=1}^T \mathbf{g}_t^\top \mathbf{w}_t - g_{n,t} \lesssim \sqrt{\log(|\mathcal{N}(\mathcal{T}_0)|) \sum_{t=1}^T (\mathbf{g}_t^\top \mathbf{w}_t - g_{n,t})^2}.$

Locally Adaptive Algorithm as expert aggregation (2/2)

Our algorithm tracks the best pruning of $\mathcal{T}_0$ i.e. the best partition $(\mathcal{X}_n)$ of $\mathcal{X}$ to recover f.

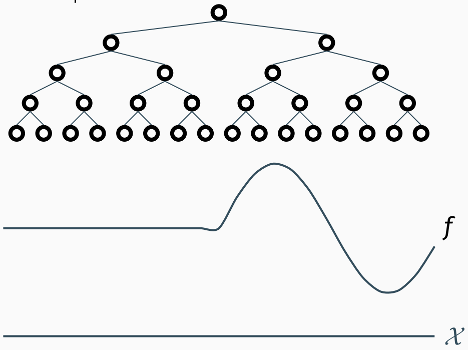
Learning with respect to a pruning of $\mathcal{T}_0$:

- given its associated partition $(\mathcal{X}_n)$ of $\mathcal{X}$,
- we define for $f \in \mathscr{C}^\alpha(L)$:

$$L_n(f) := \sup_{x,y \in \mathcal{X}_n} \frac{|f(x) - f(y)|}{\|x - y\|^\alpha} \leqslant L,$$

- and $T_n := \{1 \leqslant t \leqslant T : x_t \in \mathcal{X}_n\}, |T_n| \leqslant T$.

Example 1:



Second result: local & minimax-optimal regret

Our algorithm **optimally** competes against any **pruning** and adapts to the **local Hölder regularities** of the competitor, achieving for $\alpha \geqslant d/2$:

$$\sup_{f \in \mathscr{C}^\alpha(L)} \mathrm{Reg}_T(f) \lesssim \inf_{\text{prun}} \left\{ \sqrt{T|\text{prun}|} + \textstyle\sum_{n \in \text{prun}} 2^{-\alpha \text{level}(n)} L_n(f) \sqrt{|T_n|} \right\}.$$

- **Adaptivity** to local regularities ($L_n(f)$) w.r.t. any pruning.
- From global $O(L\sqrt{T})$ to **local** $O(\sum_n L_n \sqrt{|T_n|})$: low regret in low-variation regions!

Second result: local & minimax-optimal regret

Our algorithm **optimally** competes against any **pruning** and adapts to the **local Hölder regularities** of the competitor, achieving for $\alpha \geqslant d/2$:

$$\sup_{f \in \mathscr{C}^\alpha(L)} \mathrm{Reg}_T(f) \lesssim \inf_{\mathrm{prun}} \Big\{ \sqrt{T|\mathrm{prun}|} + \textstyle\sum_{n \in \mathrm{prun}} 2^{-\alpha \mathrm{level}(n)} L_n(f) \sqrt{|T_n|} \Big\}.$$

- **Adaptivity** to local regularities ($L_n(f)$) w.r.t. any pruning.
- From global $O(L\sqrt{T})$ to **local** $O(\sum_n L_n \sqrt{|T_n|})$: low regret in low-variation regions!

Moreover if (ℓ_t) are *exp-concave* (e.g. squared or logistic losses)

$$\sup_{f \in \mathscr{C}^\alpha(L)} \mathrm{Reg}_T(f) \lesssim \inf_{\mathrm{prun}} \Big\{ |\mathrm{prun}| + \textstyle\sum_{n \in \mathrm{prun}} 2^{-\mathrm{level}(n)} L_n(f) \sqrt{|T_n|} \Big\}.$$

- **Adaptivity** to the loss curvature.

Corollary: minimax-optimality

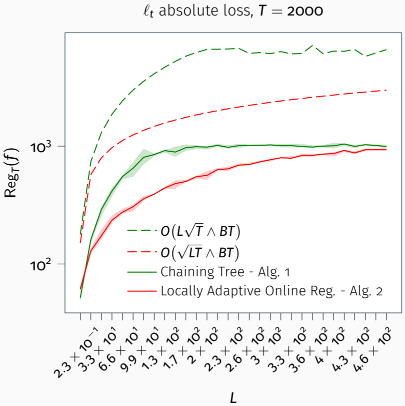
For $\alpha \geqslant d/2$, if we consider a *flat* pruning one has:

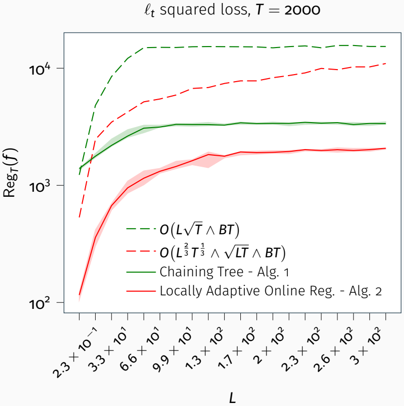
$$\sup_{f \in \mathscr{C}^\alpha(L)} \mathrm{Reg}_T(f) \lesssim \begin{cases} \left(L(f) \wedge L(f)^{\frac{d}{2\alpha}}\right)\sqrt{T}, & \text{if } (\ell_t) \text{ convex,} \\ L(f)^{\frac{d}{2\alpha}}\sqrt{T} \wedge L(f)^{\frac{2d}{2\alpha+d}} T^{\frac{d}{2\alpha+d}}, & \text{if } (\ell_t) \text{ exp-concave.} \end{cases}$$

Comparison in case $d = 1, \alpha \in [\frac{1}{2}, 1]$:

Reference	Assumptions	Regret bound
Alg. 2	(ℓ_t) exp-concave, $L > 0$ unknown (ℓ_t) convex, $L > 0$ unknown	$\min\left\{L^{\frac{1}{2\alpha}}\sqrt{T}, L^{\frac{2}{2\alpha+1}} T^{\frac{1}{2\alpha+1}}\right\}$ $L^{\frac{1}{2\alpha}}\sqrt{T}$
Kuzborskij et al. (2020)	(ℓ_t) square loss, $L > 0$ unknown, $\alpha = 1$	$\sqrt{LT}$
Hazan et al. (2007)	(ℓ_t) square loss, $L > 0$ known, $\alpha = 1$	$\sqrt{LT}$

Experiments in L: *local* adaptivity yields *smaller global* regret!





Computational feasability:

Algorithm	Time complexity	Space complexity
Alg. 1	$O(T \times \frac{1}{d} \log_2(T))$	$O(T)$
Alg. 2	$O(T \times \frac{\sqrt{T}}{d^2} \log_2^2(T))$	$O(T^{\frac{3}{2}})$

Conclusion

- We propose a parameter-free online strategy on chaining tree achieving **minimax regret**;
- A unique algorithm that both adapts to **local regularities** of the competitor and **curvature** of sequential losses;
- **First** constructive algorithm to achieve **optimal locally adaptive regret**;
- What's next? Adaptivity to (α_n) and link with multifractal analysis.

Thank you!

Questions?

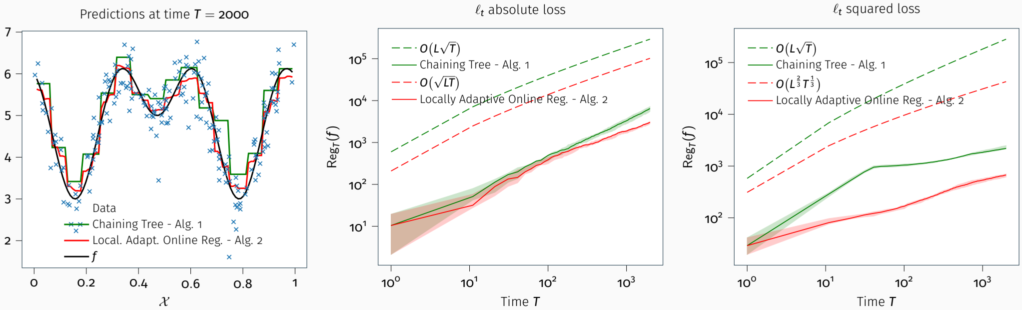
Comparison with the litterature

Ref.	Assumptions	Upper bound
[1]	(ℓ_t) exp-concave, $L > 0$ unknown	$\min\left\{\sqrt{LT}, L^{\frac{2}{3}}T^{\frac{1}{3}}\right\}$
	(ℓ_t) convex, $L > 0$ unknown	$\sqrt{LT}$
[2]	(ℓ_t) square loss, $L > 0$ unknown	$\sqrt{LT}$
[3]	(ℓ_t) absolute loss, $L > 0$ known	$L^{\frac{1}{3}}T^{\frac{2}{3}}$
	(ℓ_t) square loss, $L > 0$ known	$\sqrt{LT}$
[4]	(ℓ_t) square loss, $L = 1$ known	$T^{\frac{1}{3}}$
[5]	(ℓ_t) convex, $L = 1$ known	$\sqrt{T}$

[1] Liautaud, Gaillard, and Wintenberger, "Minimax-optimal and Locally-adaptive Online Nonparametric Regression".
[2] Kuzborskij and Cesa-Bianchi, "Locally-adaptive nonparametric online learning".
[3] Hazan, Agarwal, and Kale, "Logarithmic regret algorithms for online convex optimization".
[4] Gaillard and Gerchinovitz, "A Chaining Algorithm for Online Nonparametric Regression".
[5] Cesa-Bianchi et al., "Algorithmic chaining and the role of partial feedback in online nonparametric learning".

Experiments

Regression setting: $y_t = f(x_t) + \varepsilon_t$, where $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 0.5, f(x) = \sin(10x) + \cos(5x) + 5$, for $x \in \mathcal{X} = [0, 1]$ and $\sup_x |f'(x)| \leqslant 15 =: L$.



What about the excess-risk in batch learning?

- **Online regret bound** againt any $f \in \mathcal{F}$:

$$\frac{1}{T}\,\mathrm{Reg}_T(f) = \frac{1}{T}\sum_{t=1}^{T}(\widehat{f}_t(x_t) - y_t)^2 - \frac{1}{T}\sum_{t=1}^{T}(f(x_t) - y_t)^2 = o(1).$$

- If $\{(x_t, y_t)\}_{t=1}^{T} \overset{\text{iid}}{\sim} (X, Y)$, $\ell_t(\widehat{y}) = (\widehat{y} - y_t)^2$, excess risk of $\bar{f}_T = \frac{1}{T}\sum_{t=1}^{T}\widehat{f}_t$ is bounded as

$$\begin{aligned} \mathbb{E}\big[(\bar{f}_T(X) - Y)^2\big] - \mathbb{E}\big[(f(X) - Y)^2\big] \quad &\overset{\text{Convexity}}{\leqslant} \quad \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[(\widehat{f}_t(X) - Y)^2\big] - \mathbb{E}\big[(f(X) - Y)^2\big] \\ &= \quad \frac{1}{T}\mathbb{E}\big[\mathrm{Reg}_T(f)\big] \quad = \quad o(1). \end{aligned}$$

References

Cesa-Bianchi, Nicolò and Gábor Lugosi (2006). *Prediction, Learning, and Games.* Cambridge University Press.

Cesa-Bianchi, Nicolò et al. (2017). "Algorithmic chaining and the role of partial feedback in online nonparametric learning". In: *Conference on Learning Theory*. PMLR, pp. 465–481.

Cutkosky, Ashok and Francesco Orabona (2018). "Black-box reductions for parameter-free online learning in banach spaces". In: *Conference On Learning Theory*. PMLR, pp. 1493–1529.

Gaillard, Pierre and Sebastien Gerchinovitz (2015). "A Chaining Algorithm for Online Nonparametric Regression". In: *COLT*.

Hazan, Elad, Amit Agarwal, and Satyen Kale (2007). "Logarithmic regret algorithms for online convex optimization". In: *Machine Learning* 69.2, pp. 169–192.

Kuzborskij, Ilja and Nicolo Cesa-Bianchi (2020). "Locally-adaptive nonparametric online learning". In: *Advances in Neural Information Processing Systems* 33, pp. 1679–1689.

Liautaud, Paul, Pierre Gaillard, and Olivier Wintenberger (2024). "Minimax-optimal and Locally-adaptive Online Nonparametric Regression". In: *arXiv preprint arXiv:2410.03363*.

Mhammedi, Zakaria and Wouter M Koolen (2020). "Lipschitz and comparator-norm adaptivity in online learning". In: *Conference on Learning Theory*. PMLR, pp. 2858–2887.

Orabona, Francesco and Dávid Pál (2016). "Coin betting and parameter-free online learning". In: *Advances in Neural Information Processing Systems* 29.

Rakhlin, Alexander and Karthik Sridharan (2014). "Online non-parametric regression". In: *Conference on Learning Theory*. PMLR, pp. 1232–1264.

— (2015). "Online nonparametric regression with general loss functions". In: *arXiv preprint arXiv:1501.06598*.

Zinkevich, Martin (2003). "Online convex programming and generalized infinitesimal gradient ascent". In: *Proceedings of the 20th international conference on machine learning (icml-03)*, pp. 928–936.